**HANDWRITTEN CALCULATOR USING CNN**

A project report submitted in partial fulfillment of the requirements for the award of the degree of

**Master of Computer Applications**

**in**

**Computer Applications**

**By**

**HRISHIKANT MEHTA (205120036)**

****

**DEPARTMENT OF COMPUTER APPLICATIONS**

**NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI 620015**

**DECEMBER 2022**

**BONAFIDE CERTIFICATE**

This is to certify that the project **“Handwritten Calculator using CNN”** is a project work successfully done by

Hrishikant Mehta (205120036)

in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications from National Institute of Technology, Tiruchirappalli, during the academic year 2021-2022 (5th Semester – CA749 Mini Project Work).

Dr. S.R. Balasundaram Dr. P.J.A. Alphonse

Project Guide Head of the Department

Project viva-voce held on …………………………….

**Internal Examiner** **External Examiner**

ABSTRACT

Handwritten character or digit recognition has gained huge attention in the last decade. We see many applications using this in our daily life, like OCR (optical character recognition) apps, smart keyboards where you can type characters by drawing them, many mathematical applications are also emerging using character recognition techniques, where we can solve handwritten equations, etc. But till now also it is not developed enough to show very good accuracy in real life data. In this project, I have tried to make a simple arithmetic expression solver using CNN (Convolutional neural network) model. I have also made a web application out of it, where it can be used by normal users by directly writing the expression and getting the result. CNN works the best in image data and recognition of objects in images. I have used some state-of-the-art techniques also in data pre-processing, and model architecture. The accuracy of the model was coming as 99.09% while training and with the dataset’s test data itself. And with the technique used for extracting the characters and symbols from the expression and feeding to the model, combined was giving excellent results in real time data (handwritten expression).

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to all who have helped me in doing this project. Special thanks to my project guide Dr. B. Balaji who has guided me while making this project via continuous review and suggestions for changes and other things. My mini project lab guide also helped a lot mainly in coding part of this project, I express my thanks to him also.

Apart from this I also thank my parents who have supported me in each point in my life, friends with whom I discussed and gained some new perspective for making this project, YouTube tutor from which I have learned deep learning and machine learning in Hindi language, he does explanation of every topic in very easy, so that anyone can understand.

Finally, I would like to thank God, for letting me through all difficulties I faced throughout, when no one was there. I will keep trusting you in my future as well.

TABLE OF CONTENTS

| **Topic** | **Page No** |
| --- | --- |
| Introduction | 4 |
| Related Work | 5 |
| Dataset Overview |  |
| Algorithms used |  |
| Architecture |  |
| Experimental Setup |  |
| Experimental Result |  |
| Conclusions and Future Work |  |
| References |  |

INTRODUCTION

Handwritten Calculator is a kind of calculator where user can write the expression by hand and the calculator will automatically evaluate the expression after recognizing the digits and symbols. Nowadays digital writing is emerging very fast, for example, in virtual courses tutor uses big touch screen for writing, people make notes in their mobiles, it can be handwritten also. Many applications are also emerging in the field of mathematics for solving the equations, expressions etc. This project can act as crucial component in these kinds of applications.

In the Artificial Intelligence domain, we have machine learning techniques also to do classification, it can be used for image classification also. But after the emergence of more computing power with good hardware like GPUs, many deep learning model came into picture. CNN (Convolution neural network) is one of the deep learning models which works very well with image data. It also does automatic feature extraction form the image for visual recognition.

The purpose of this project to make a calculator which can solve handwritten expressions. This can be in the form of web application or mobile application also; in this project I have shown it via web app. The application uses the CNN model in background to recognize the digits and symbols.

The major work and novelty of this project is to do pre-processing in a way so it can show very good accuracy in real data, a slightly different model than traditional model for digit recognition so that loss of information before feeding the image into model doesn’t make much effect, and making of the calculator for the application, it also includes deciding how the digits and symbols are extracted from expression for its valid classification.

RELATED WORK

Many research and studies has been done in the field of handwritten digit classification, with very good accuracy. Some of them are listed below:

1. Hybrid CNN-SVM Classifier for Handwritten Digit Recognition Savita Ahlawata , Amit Choudhary b [[1].](https://www.sciencedirect.com/science/article/pii/S1877050920307754)
2. In 1995 SVM was used for first time handwritten character recognition OCR.

Cortes, C. and Vapnik V. (1995) “Support vector networks”, Machine Learning, 20.

1. Now a days deep learning model is used with popular MNIST dataset, which contains 70000 images of different digits. Most of the experiments achieved accuracy more than 98% or 99%.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009) “What is the best multi-stage architecture for object recognition?” In: Proceedings of IEEE 12th International Conference on Computer Vision (ICCV) 2146-2153.

1. The highest recognition accuracy achieved was 99.73%, using ensemble of multiple CNN.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, M. L., Schmidhuber, J., Flexible, High-Performance Convolutional Neural Networks for Image Classification, In proceedings of Twenty-Second International Joint Conference on Artificial Intelligence, 1237-1242, 2011.

This project is also motivated by these papers, along with little bit difference in model and dataset used here. The model has also made keeping the application centric requirements like my input in always taken by digital handwriting, which is already black and white, and other changes are also discussed in upcoming topics.

DATASET OVERVIEW

Dataset used in this project is “[Handwritten Math Symbols](https://www.kaggle.com/datasets/sagyamthapa/handwritten-math-symbols)” from Kaggle which contains almost 500 images of each digit.

From this dataset I have used only 14 type of images which are digits from 0 to 9 and 4 arithmetic symbols. In total it contains almost 7000 images.

In this project traditional MNIST dataset was not used because it doesn’t contain the images for operator symbols, and it contains images of resolution 28\*28 pixel only.

This dataset contains images in high resolution. This gives the benefit of preprocessing the data in customized way. Using this I was able to make my model which takes the image in 40\*40 pixel resolution rather than traditional 28\*28 pixel format.

Sample images from the dataset is given below:

Shape, arrow

Description automatically generated

ALGORITHMS USED

There are many algorithms used in this project. Some of them are listed below:

1. Preprocessing
   1. Edge detection using OpenCV:

OpenCV contains findContour function to detect the edges in an image. An edge is nothing but sudden change in pixel value.

* 1. Bounding Rectangle:

OpenCV also contains a function to find bounding rectangle of an object inside image. It takes the contour of that object as its parameter.

Edge detection and bounding rectangle is used for cropping the relevant part only from the image. Here relevant part means only keeping those area which contains the stroke of the digit. There are many columns and rows which are completely blank. We remove those pixels.

1. Model Training and Prediction
   1. CNN:

CNN is not a model, but it is a neural network which has been used here. Along with this model different functions are used for hyperparameters like activation function (relu and SoftMax), optimizer (Adam) and loss function (categorical cross entropy).

* 1. This all has been done in python language with some libraries like keras, TensorFlow, NumPy, pandas etc.

ARCHITECTURE

The architecture of the CNN model used here is as follows:

Convolution layers:

Chart, waterfall chart

Description automatically generated

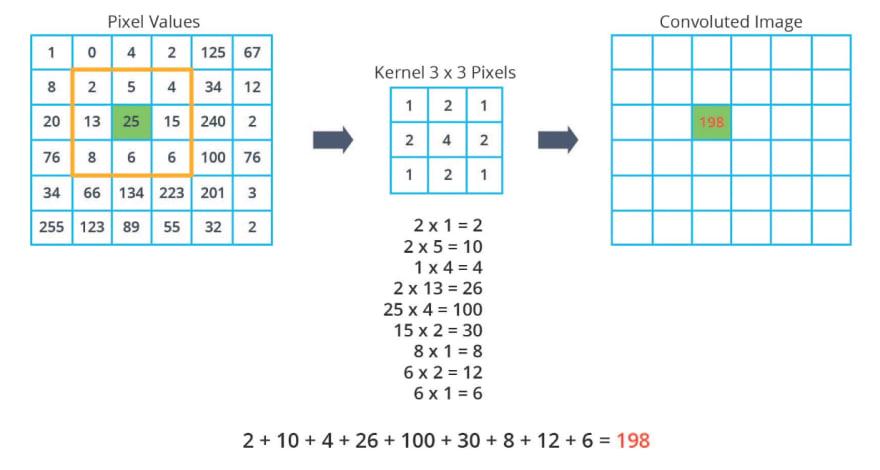
Fully Connected layers:

Diagram, schematic

Description automatically generated

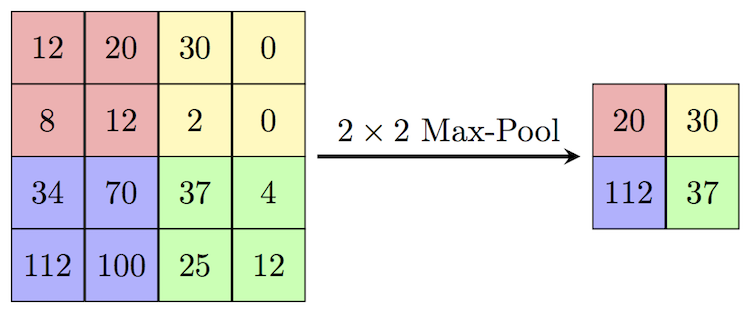
1. Convolution Layers

In Convolutional neural networks, convolution layer is the first set of layers by which image goes through. It is used to detect features from the image. In the initial layers, it detects simple edges like slanted lines at different angles, then in further layers, it detects complex features like polygons etc. A kernel is a matrix, which convolves though the image and gets values, after convolution, the output is called feature map.



1. Pooling layers

Pooling layers are also crucial component in Convolutional neural networks. It is used to reduce the computational complexity of the model. This also contains a matrix which convolves through the image and extract only the max, min, or average value according to the different pooling that is being used such as max pooling, min pooling, average pooling, etc. It reduces the dimension of the model by keeping the information about dominant features in rotation or translational aspect of the image.



1. Flatten Layer

Flatten layer make the output from the convolutional layer to one dimension. After flatten layer, we can add fully connected layers which is nothing but general artificial neural network.

1. Fully connected layers

These layers are added after the convolutional layer and produces final output. In the above model, the fully connected layers contain one hidden layer and one output layer containing 14 output nodes.

EXPERIMENTAL SETUP

The whole experiment was divided into following steps:

1. Collecting the dataset
2. Preprocessing
3. Building and training CNN model
4. Character and operator symbol separation from given expression as image
5. Prediction of each element (digits and operators)
6. Creating the calculator
7. Deployment in web application

All steps up to creating the calculator was done in google colab only.

1. Dataset collection

Already discussed in Dataset overview part

1. Preprocessing

The images were of high resolution, and the written characters were also not in the center in each image, so to make them similar and in center, following steps were performed:

1. Take original image.
2. Crop using bounding rectangle.
3. Scale width or height (whichever is maximum) to 40 pixels.
4. Pad empty pixels row or column on both side of height or width (whichever is minimum).
5. Reshape and save these 40\*40 images.

It will look something like this:

Diagram

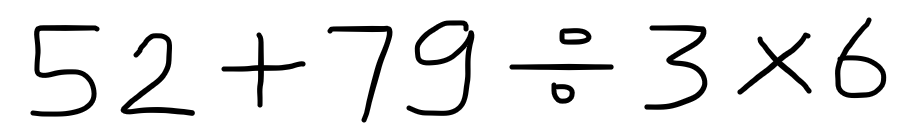
Description automatically generated

1. Building and training CNN model

For this I have used keras library which works on the top of TensorFlow library. Model takes the 40\*40 image as its input.

1. Character and operator symbol separation from given expression as image

This step is the crucial step where an image containing the expression can be like this:



For this image we find the same bounding rectangle for each character and after cropping each one of them, we do the same thing as the preprocessing.

A picture containing diagram

Description automatically generated

1. Prediction of each element (digits and operators)

After separating each element, we feed it to the model, and model predicts it.

1. Creating the calculator

For the calculator part, we make an expression as string by the prediction given by our model, and using eval function of python, we evaluate it.

52 + 79 / 3 \* 6 = 210.0

1. Deployment in web application

For the deployment part. First model was saved and being used in the backend of web app. I have used streamlit library of python. Streamlit is python library by which we can make quick web application using python code only. Although it causes the web app to be heavy but, for short time it is okay. Screenshots can be seen in Experimental results part.

EXPERIMENTAL RESULTS

While model training the accuracy achieved was 99.09%. The number of epochs for training was 10, with adam optimizer, and loss function as categorical cross entropy.

And, with the real data via web application the accuracy was excellent. After multiple testing, it was found that almost 1 or 2 times, one of the characters in the expression were predicted wrongly out of 20 times.

Screenshot:

Text

Description automatically generated with medium confidence

CONCLUSION ANS FUTURE WORK

In this project, I have implemented a simple arithmetic expression solver as a calculator using CNN model. The accuracy achieved was also very good. I have also made some changes in the traditional model for digit classification such as using 40\*40 pixel as input images. This project can be extended for accepting more characters like parentheses, exponent operators, decimal points, etc. We can also extend this for simple equation solver by introducing few variable symbols like x and y, etc.

REFERENCES

1. Hybrid CNN-SVM Classifier for Handwritten Digit Recognition Savita Ahlawata , Amit Choudhary b [[1].](https://www.sciencedirect.com/science/article/pii/S1877050920307754)
2. Cortes, C. and Vapnik V. (1995) “Support vector networks”, Machine Learning, 20.
3. Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009) “What is the best multi-stage architecture for object recognition?” In: Proceedings of IEEE 12th International Conference on Computer Vision (ICCV) 2146-2153.
4. Ciresan, D. C., Meier, U., Masci, J., Gambardella, M. L., Schmidhuber, J., Flexible, High-Performance Convolutional Neural Networks for Image Classification, In proceedings of Twenty-Second International Joint Conference on Artificial Intelligence, 1237-1242, 2011.
5. 100 days of deep learning in youtube: <https://youtube.com/playlist?list=PLKnIA16_RmvYuZauWaPlRTC54KxSNLtNn>
6. Introduction to CNN Keras:

<https://www.kaggle.com/code/yassineghouzam/introduction-to-cnn-keras-0-997-top-6/notebook>